**DR CATALYST EHR (WebIMS)**

A PROJECT REPORT

*Submitted by*

**Om Siddhapura (19CP048)**

*In partial fulfillment for the award of the degree of*

**B. TECH. in COMPUTER ENGINEERING**

**4CP33: Full Semester External Project (FSEP)**



**BIRLA VISHVAKARMA MAHAVIDYALAYA**

**(ENGINEERING COLLEGE)**

*(An Autonomous Institution)*

**VALLABH VIDYANAGAR**

*Affiliated to*



**GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD**

**Academic Year: 2022 – 2023**

BVM ENGINEERING COLLEGE, VALLABH VIDYANAGAR-388120

# **APPROVAL SHEET**

The project work entitled “**Dr Catalyst EHR**" carried out by **Om Siddhapura (19CP048)** is approved for the submission in the course **4CP33, Full Semester External Project** for the partial fulfillment for the award of the degree of B. Tech. in Computer Engineering.

Date:

Place:

Signatures of Examiners:

(Names and Affiliations)

# **CERTIFICATE**

This is to certify that Project Work embodied in this project report titled “**Dr Catalyst EHR** " was carried out by **Om Siddhapura (19CP048)** under the course **4CP33, Full Semester External Project** for the partial fulfillment for the award of the degree of B. Tech. in Computer Engineering. Followings are the supervisors at the student:

Date:

Place:

| Mr. Manish Lakhara | Prof. Prashant B. Swadas | Dr. N.M. Patel |
| --- | --- | --- |
| Team-Lead (Front-end) at WebIMS,  Meditab Software (India) Pvt. Ltd.,  Ahmedabad | Associate Professor,  Computer Engineering Department,  BVM Engineering College | Professor,  Computer Engineering Department,  BVM Engineering College |

| Dr. Darshak G Thakore |
| --- |
| Prof. & Head,  Computer Engineering Department,  BVM Engineering College |

COMPUTER ENGINEERING DEPARTMENT, BVM ENGINEERING COLLEGE, VALLABH VIDYANAGAR-388120

**CERTIFICATE**



**DECLARATION OF ORIGINALITY**

I hereby certify that I am the sole author of this report under the course 4CP33 (Full Semester External Project) and that neither any part thereof nor the whole of the report has been submitted for a degree to any other University or Institution.

I certify that, to the best of my knowledge, the current report does not infringe upon anyone’s copyright nor does it violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in my report, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in the current report and have included copies of such copyright clearances to the appendix of this report.

I declare that this is a true copy of report, including any final revisions, as approved by the report review committee.

I have checked write-up of the present report using anti-plagiarism database and it is in permissible limit. However, at any time in future, in case of any complaint pertaining of plagiarism, I am the sole responsible person for the same. I also understand that, in case of such complaints of plagiarism, as per the University Grants Commission (UGC) norms, the University can even revoke the degree conferred to the student submitting such a plagiarized report.

Date:

Institute Code: 007

Institute Name: Birla Vishvakarma Mahavidyalaya (BVM) Engineering College

|  |
| --- |
| Om Siddhapura |
| 19CP048 |

# **ACKNOWLEDGEMENT**

I would like to present the project with immense pleasure and commitment. The development of this project has given me so many opportunities to think, explore and get to know about the Web Development domain.

Every work that one completes successfully stands on the constant encouragement, good will and support of the people around. I hereby avail this opportunity to express our gratitude to number of people who extended their valuable time, full support and cooperation in developing the project.

I express deep sense of gratitude towards our Head of the Computer Department, **Dr. Darshak G Thakore** and project guides to **Prof. Prashant B. Swadas** (internal guide) and **Dr. Narendra M. Patel** (internal guide) as well as **Prof. Mosin I. Hasan** and **Mr. Manish Lakhara** (external guide) for the support during the whole session of study and development. It is because of them, that I was prompted to do hard work, adopting new technologies.

Furthermore, I am grateful to all the people who kept supporting me and provided a positive environment to work with.

| Om Siddhapura |
| --- |
| 19CP048 |

# **Plagiarism Report**

# Average Plagiarism: 7.6%

# Tool used: <https://www.duplichecker.com/>

# 

# 

# **Abstract**

Main idea of EHR (Electronic Health Management) system projects is to use the software application for clinics and hospitals to store patient data electronically and to easily maintain patient data and billing system. In the current medical service systems and patient information is stored in the form of manual records. It is tedious process to manually maintain patient data record for long time, and retrieve them. The US health care work flow is very inefficient and it is hard for patients to retrieve their data and communicate with health care provider. Dr.Catalyst EHR (WebIMS) provides one stop solution for all these problems. Dr.Catalyst EHR provides functionality such as patient scheduling, charting, e- prescribing, billing and claims management, patient portals, and reporting and analytics. The software is highly customizable, with different modules and configurations available to meet the specific needs of different types of healthcare providers and it complies HIPAA, which ensures healthcare providers that their patient data is secure and that they are meeting regulatory requirements. One of the major advantages of this software is that it allows healthcare providers to access patient data and clinical workflows from anywhere, at any time. Smaller medical practices, those that need to work remotely, or those who need to operate from numerous locations may find this to be very helpful.

# **TABLE OF CONTENTS**

**CERTIFICATE.......................................................................................................................... III**

**APPROVAL SHEET.................................................................................................................. II**

**DECLARATION OF ORIGINALITY..................................................................................... V**

**ACKNOWLEDGEMENT.........................................................................................................VI**

**PLAGIARISM REPORT........................................................................................................ VII**

**ABSTRACT............................................................................................................................. VIII**

**TABLE OF CONTENTS.......................................................................................................... IX**

**LIST OF FIGURES………………………………………………………………..…….….. XII**

**LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE................................................................................................................ XIII**

**CHAPTER 1: INTRODUCTION ............................................................................................... 1**

1.1 Project summary....................................................................................................................... 1

1.2 Purpose...................................................................................................................................... 1

1.3 Scope......................................................................................................................................... 1

1.4 Objective................................................................................................................................... 1

**CHAPTER 2: Related Work and Background.......................................................................... 3**

2.1 Functional requirements........................................................................................................... 3

2.1.1 Patient management..................................................................................................... 3

2.1.2 Appointment scheduling.............................................................................................. 3

2.1.3 Billing and claims management..................................................................................... 3

2.1.4 Clinical workflows....................................................................................................... 3

2.1.5 Prescription management.............................................................................................. 3

2.1.6 Reporting and analytics:…………………………………………………………….. 3

2.2 Nonfunctional requirements………………………………………………………………...4

2.2.1 Performance…………………………………………………………………..…….4

2.2.2 Usability…………………………………………………………………...……….. 4

2.2.3 Availability……………………..…………………………………………………. 4

2.2.4 Maintainability……………………………………………….……………………... 4

2.2.5 Portability:………………………………………………………………………….. 4

2.2.6 Manageability……………………………………………….…………………….... 4

2.2.7 Data integrity: ……………………………………………………………………... 4

2.3 Technical requirements………………...…………………………………………...….…... 5

2.3.1 Development Environment...................................................................................... 5

2.3.2 Execution Environment ........................................................................................... 5

2.4 Technologies…………………………………………………………………..…….………. 6

2.4.1 Java Script…………………………………………………………………….…… 6

2.4.2 .NET API…………………………………………………………………………. .6

2.4.3 Postgresql………………………………………………………….…….…............ 7

2.4.4 AWS………………………………………………………………………..….…... 8

2.4.5 Postman……………………………………………………………………...…….. 8

2.4.6 Angular…………………………………………………………………...….…….. 9

**CHAPTER 3: MODELING AND DESIGN ………………………………………………… 10**

3.1 ER diagram……………………………………………………………….………………… 11

3.2 Use case diagram…………………………………..…………...…………………………... 12

3.3 State Diagram…………………………………………………………...………………….. 13

3.4 Activity Diagram………………………………………….………………………………... 14

3.4.1 Patient………………………………………………...…………………………... 14

3.4.2 Admin……………………………….………………………………………......... 15

3.5 Data flow Diagram…………………………………………………………………….…… 16

3.6 Sequence Diagram………………………………………………………………….……… 17

**CHAPTER 4: IMPLEMENTATION...................................................................................... 18**

4.1 Demographics module…………….………………………….………….………………… 18

4.1.1 Front-end………………….……………………………………………………… 18

4.1.2 Back-end………………………………………………………..………………... 19

4.1.2.1 Operational Tables……………………………………………………... 19

4.1.2.2 APIs…………………………………………………………................. 20

4.1.2.2.1 Create patient with allergy …………………………………….. 20

4.1.2.2.2 Update patient with allergy…………………………………….. 21

4.1.2.2.3 Get patient by id……………………………….......................... 22

4.1.2.2.4 Delete patient with allergy……………………………................ 23

4.2 Check-In/Check Out module…………….………………………….………….………….. 24

4.2.1 Front-end………………….……………………………………………………… 24

4.3 Employee Availability module…………….………………………….………….………… 26

4.3.1 Front-end………………….……………………………………………………… 26

**CHAPTER 5 : CONCLUSION AND FUTURE SCOPE ……………………...………28**

5.1 Conclusion….………….………………………….………….…………………………….. 28

5.2 Limitations…………….………………………….………….……………………………... 28

5.3 Future scope…………….………………………….………….……………………………. 29

REFERENCES…………………………………………………………………...…………..… 30

# **List of Figures**

| **Figure**  **Number** | **Figure Description** | **Page**  **Number** |
| --- | --- | --- |
| 2.1 | Java Script | 6 |
| 2.2 | .NET API | 7 |
| 2.3 | PostgreSQL | 7 |
| 2.4 | AWS | 8 |
| 2.5 | Postman | 9 |
| 2.6 | Angular | 9 |
| 3.1 | ER diagram | 11 |
| 3.2 | Usecasediagram | 12 |
| 3.3 | Statediagram | 13 |
| 3.4 | Activitydiagram\_patient | 14 |
| 3.5 | Activitydiagram\_admin | 15 |
| 3.6 | Dataflowdiagram | 16 |
| 3.7 | Sequencediagram | 17 |
| 4.1 | Demographicswebview(1) | 18 |
| 4.2 | Demographicswebview(2) | 19 |
| 4.3 | operationaltables | 20 |
| 4.4 | Create patient with allergy | 21 |
| 4.5 | Update patient with allergy | 22 |
| 4.6 | Get patient details by id | 23 |
| 4.7 | Delete patient with allergy | 24 |
| 4.8 | CheckIn/CheckOut screen | 25 |
| 4.9 | Appointment Details | 25 |
| 4.10 | Update Appointment | 26 |
| 4.11 | Employee Availability | 27 |
| 4.12 | Patient Reminders | 27 |

# **List of Symbols, Abbreviations and Nomenclature**

* EHR - Electronic Health Record
* HIPAA - Health Insurance Portability and Accountability Act
* IMS – Intelligent Medical Software
* MA – Medical Assistant
* AWS – Amazon Web Services
* API – Application Programming Interface
* HTTPS - Hypertext Transfer Protocol
* ER - Entity Relationship
* UI - User Interface
* RAM - Random-access memory
* MB - MegaBytes
* GB - GigaBytes

# **Chapter 1**

# **Introduction**

**1.1 PROJECT SUMMARY**

Dr.Catalyst EHR is Web based software. It makes the life of providers (doctors) much easier by converting file-based systems to software-based systems. This is done by registering the entire patient in the IMS software and managing all the patient related information in the patient master module. Medical Insurance is mandatory in the US. To manage the patient insurance our system uses the patient insurance module to keep the records of the patient insurance and help to track all the claims. Thereafter, as the number of patients is huge, a search screen module is available in our software. This makes it easier for the user to search the patients, insurances, providers, etc. Instruction master is used to provide the facility of jotting down predefined instructions for a particular action. Authorization tracking module is used to track the claims that need to be sent to the insurance company.

**1.2 PURPOSE**

IMS systems would be given to the providers (doctors) as well as the nurse practitioners or the receptionists at the hospital or the clinic. The front desk personnel will use the software as well as the provider will be using the software. Only difference is that both of them would be using different modules according to their needs and according to the accessibility to the modules. The provider would input the vitals and the disease related information into the module and the front personnel would enter the data regarding the patient. Here, the purpose of the IMS in the clinic is to enter the medical related data and to retrieve the data back whenever required**.**

**1.3 SCOPE**

Intelligent Medical Software will provide a platform for the providers and the clinical staff to concentrate on their main work, which is to give maximum time to the practice rather than concentrating on the paperwork. The advantages of Intelligent Medical Software in the clinical setting are important. According to a published report, 81% of physicians (providers) said they believe IMS improves access to clinical data. More than two-thirds said an IMS can actually improve patient care.

**1.4 OBJECTIVE**

* The main objective of the project on medical office web application is to manage the details of patients, Medicine, Medicine Company, Facility, Medicine Stock, Sells, Super Bill, Patient Insurance, Provider etc. It manages all the information about customer, supplier, sells, patient. The project is totally built at administrative end and this only the clinic administrator is guaranteed the access. Because in the project there is one module of security rights to percale screens.
* It is a mainly use for billing purpose, The medical billing process is a process that involves a third-party payer, which can be an insurance company or the patient. Medical billing results in claims.
* The claims are billing invoices for medical services rendered to patients. The entire procedure involved in this is known as the billing cycle sometimes referred to as Revenue Cycle Management. A patient can do appointment remotely, and take insurance from any medical company.

Functionalities provides by Intelligent electronical medical Office are as follows:

* Provides the searching facilities based on various factors. Such as Patient, Medicine Company, Medicine Stock, Sells and Insurance Company.It also manages the supplier details online for Medicine stock details, sells details, customer. Then manage the billing provider details, visit note status, appointment details of patient , room allocation to patient , patient case, patient, family history, patient insurance claims, Patient prescription, patient receipts, etc.
* Shows the information and description of the patient, clinic.
* Integration of all records of sells,

**Chapter 2**

**Related Work and Background.**

* 1. **Functional requirements**
     1. **Patient management:** WebIMS should allow healthcare providers to create, manage, and access patient records, including demographics, medical history, clinical notes, and appointment history.
     2. **Appointment scheduling:** WebIMS should allow healthcare providers to schedule, manage, and access appointments, including automated appointment reminders.
     3. **Billing and claims management:** WebIMS should allow healthcare providers to create, manage, and access claims, process payments, and generate reports related to billing and claims management.
     4. **Clinical workflows:** WebIMS should allow healthcare providers to create, manage, and access clinical workflows, such as order sets and protocols, to improve clinical processes and patient outcomes.
     5. **Prescription management:** WebIMS should allow healthcare providers to create, manage, and access prescriptions, including electronic prescribing and refill requests.
     6. **Reporting and analytics:** WebIMS should allow healthcare providers to generate reports and analyze data related to patient care and practice performance.
  2. **Nonfunctional requirements**
     1. **Performance**: This includes requirements related to the speed, scalability, and responsiveness of the system.
     2. **Usability**: This includes requirements related to the ease of use and understandability of the system for the end-users.
     3. **Availability**: This includes that the features and functionalities are always available to users at any point of time.
     4. **Maintainability**: This includes requirements related to the ease of maintaining the system, including testing, debugging, and modifying the system.
     5. **Portability**: This includes requirements related to the ability of the system to be easily transferred to different hardware or software environments.
     6. **Manageability**: This feature is defined as the ability to control a system efficiently and keep it fully operational.
     7. **Data integrity**: Data integrity refers to maintaining and assuring data accuracy and consistency over its entire lifecycle.
  3. **Technical requirements**

### **2.3.1 Development Environment**

Software Requirement

* + - * Angular 11
      * .NET
      * PostgreSQL
      * Visual Studio
      * VS Code (1.77.1)
      * Postman
      * Any compatible browser

Hardware Requirement

* + - * 8 GB RAM
      * 256 GB SSD
      * Processor: Intel i5 10th generation
      * Mouse: Any Compatible
      * Keyboard: Any Compatible
      * Display: Any Compatible

### **2.3.2 Execution Environment**

Software Requirement

* + - * Any Browser.

Hardware Requirement

* + - * RAM – 2GB
      * Device with decent internet connectivity
      * Mouse: Any Compatible
      * Display: Any Compatible

**2.4 Technologies**

**2.4.1 Java Script [01]**

JavaScript is an object-based scripting language which is lightweight and cross-platform. JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser..

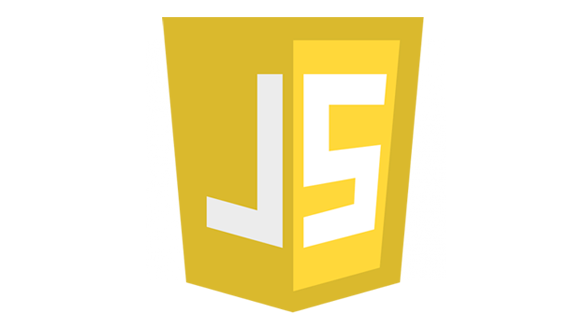


Fig. 2.1 Java Script [02]

**2.4.2 .NET API [03]**

The ASP.NET Web API is an extensible framework for building HTTP based services that can be accessed in different applications on different platforms such as web, windows, mobile etc. It works more or less the same way as ASP.NET MVC web application except that it sends data as a response instead of html view. It is like a webservice or WCF service but the exception is that it only supports HTTP protocol.

ASP.NET Web API Characteristics

ASP.NET Web API is an ideal platform for building RESTful services.

ASP.NET Web API is built on top of ASP.NET and supports ASP.NET request/response pipeline

ASP.NET Web API maps HTTP verbs to method names.

ASP.NET Web API supports different formats of response data. Built-in support for JSON, XML, BSON format.



Fig. 2.2 .NET API [04]

**2.4.3 Postgresql [05]**

PostgreSQL (pronounced as post-gress-Q-L) is an open source relational database management system (DBMS) developed by a worldwide team of volunteers. PostgreSQL is not controlled by any corporation or other private entity and the source code is available free of charge.

PostgreSQL supports a large part of the SQL standard and offers many modern features including the following −

Complex SQL queries

SQL Sub-selects

Foreign keys

Trigger

Views

Transaction

**2.4.4 AWS [07]**

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay-as-you-go basis. Often times, clients will use this in combination with autoscaling (a process that allows a client to use more compute in times of high application usage, and then scale down to reduce costs when there is less traffic). These cloud computing web services provide various services related to networking, compute, storage, middleware, IOT and other processing capacity, as well as software tools via AWS server farms. This frees clients from managing, scaling, and patching hardware and operating systems.



Fig. 2.4 AWS [08]

**2.4.5 Postman [09]**

Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.

While using Postman, for testing purposes, one doesn't need to write any HTTP client network code. Instead, we build test suites called collections and let Postman interact with the API.

In this tool, nearly any functionality that any developer may need is embedded. This tool has the ability to make various types of HTTP requests like GET, POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python.



Fig. 2.5 Postman [10]

**2.4.6 Angular [11]**

Angular is a development platform and framework for building single-page client applications using HTML and TypeScript. Angular is built using TypeScript.

Angular has many improvements and innovations over its predecessor, AngularJS. It is easy to learn and develop enterprise-scale applications. You can build extendable, maintainable, testable, and standardized applications using Angular.

Angular provides all the core functionality that you need to develop a client application. It comes bundled with a router module, an HTTP module, a forms API, etc. We can use it to build amazing client-side applications.



Fig. 2.6 Angular [12]

**Chapter 3**

**Modeling and Design**

For better visualization of implemented methodology, software design diagrams were constructed which provided a glimpse of the flow of control as well as the data in our project. The following diagrams proved useful information for a better understanding of this project.

* The ER diagram illustrates the entities that are involved in the application and also the type of data information that they are having.
* The Use case diagram tells about the user and their possible set of use cases. What different kinds of features the user can hold.
* The State Diagram indicates the possible transition between different states.
* The activity diagram will show all the activities carried out by the user to use the system efficiently and effectively for their own benefit.
* Data flow presents how the flow of the data in the application goes.
* The Sequence Diagram will show the simplest form of interaction between the user to the system.

**3.1 ER diagram(System Generated)**

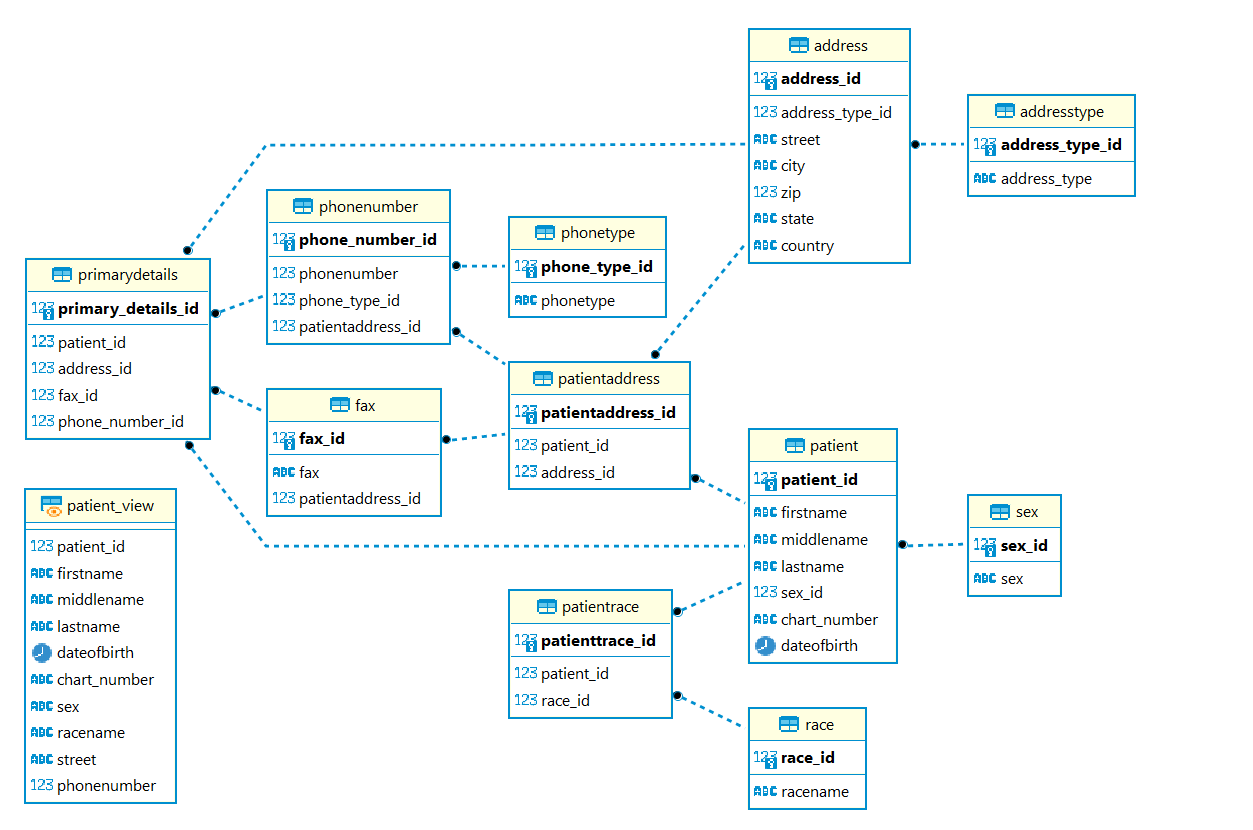


Fig. 3.1 ER diagram

The figure 3.1 shows the ER diagram of the demographics module which contains the entities like patient, patientrace, race, sex, address, patientaddress, fax, phonenumber, primarydetails, phonetype etc. Also a view containing information about the patient and corresponding address, phone ,fax details.

**3.2 Use case diagram**

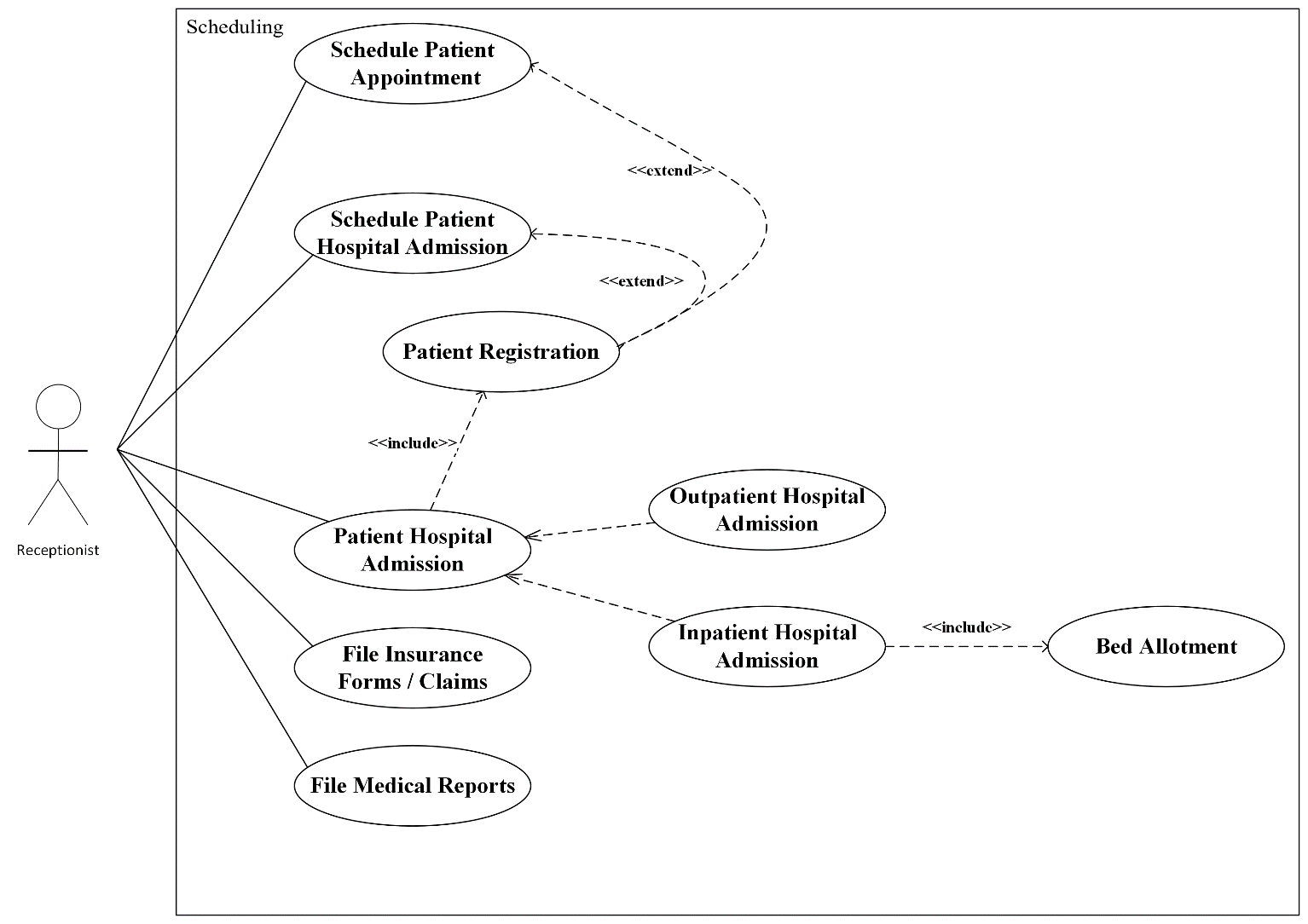


Fig. 3.2 Usecasediagram

The diagram 3.2 contains the view of the receptionist and corresponding possible use cases with it. The receptionist can schedule appointments for patient and hospital admission. Along with that, patients can be registered from admission. The receptionist is responsible for filling medical reports , file insurance. Also the management of the checkin/checkout can be controlled along with resource allotment like bed, rooms etc.

**3.3 State Diagram**

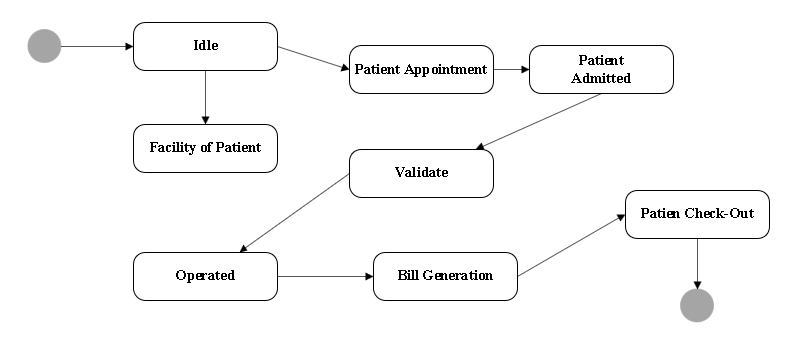


Fig. 3.3 Statediagram

The figure 3.3 presents the overall state transitions for a patient during a clinic visit. First patient would take an appointment after that he/she will be added to the system. After validation the patient will be treated by the provider and at the end the bill will be generated as per the treatment and the patient will be checked out.

**3.4 Activity Diagram**

**3.4.1 Patient**

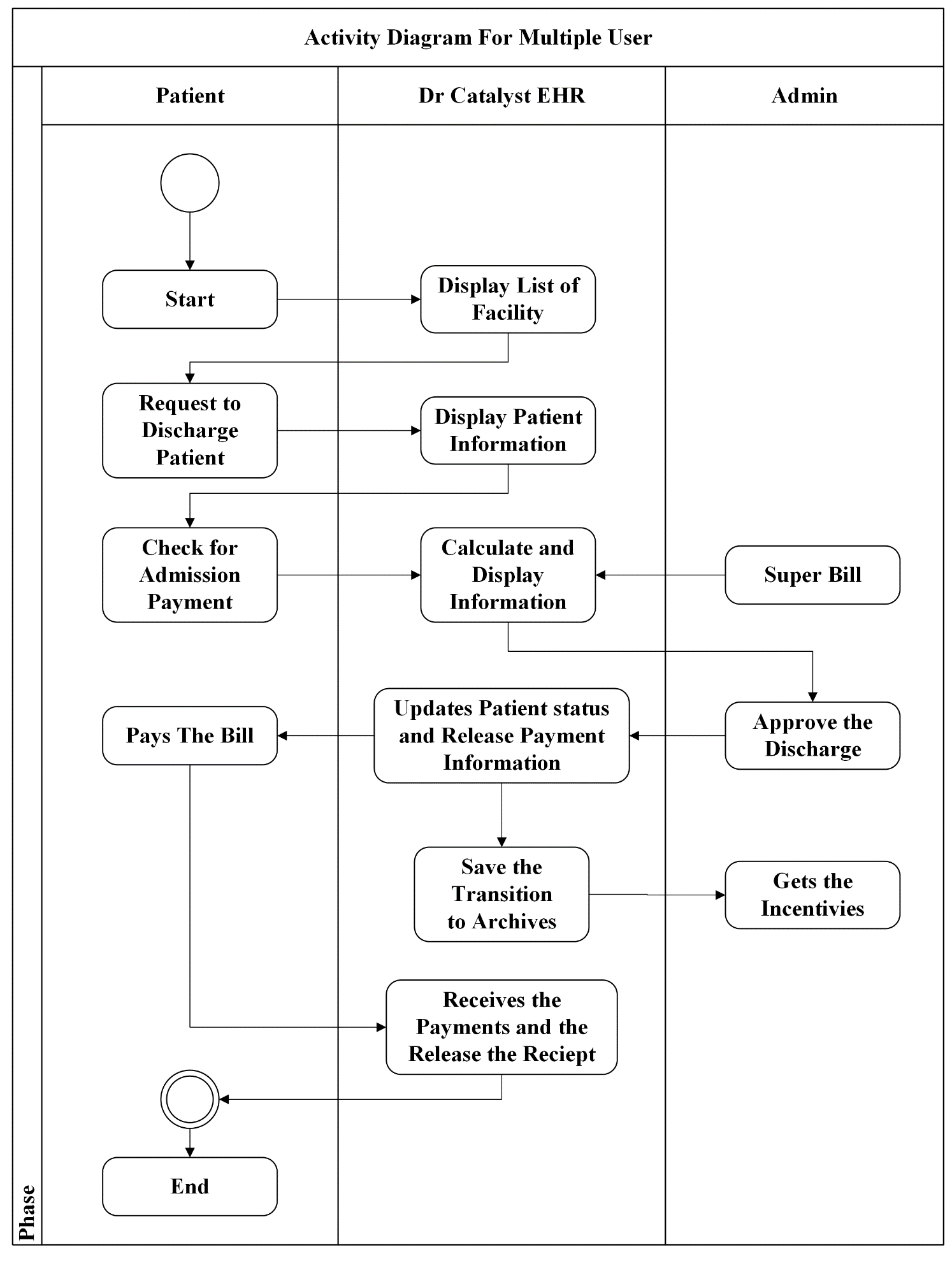
****

Fig. 3.4 Activitydiagram\_patient

The figure 3.4 indicates activities associated with the patient along with Dr Catalyst EHR system and admin. Once a patient comes to the clinic for an appointment, the receptionist can be able to see all the facilities for the patient . After the treatment patient can request to get discharged once the details regarding bills and other things are figured out from the admin super bill the admin will approve the discharge request of the patient. If the patient will pay the bill the system will generate receipts and the admin will get incentives from that. At the end the patient will be checked out.

**3.4.2 Admin**

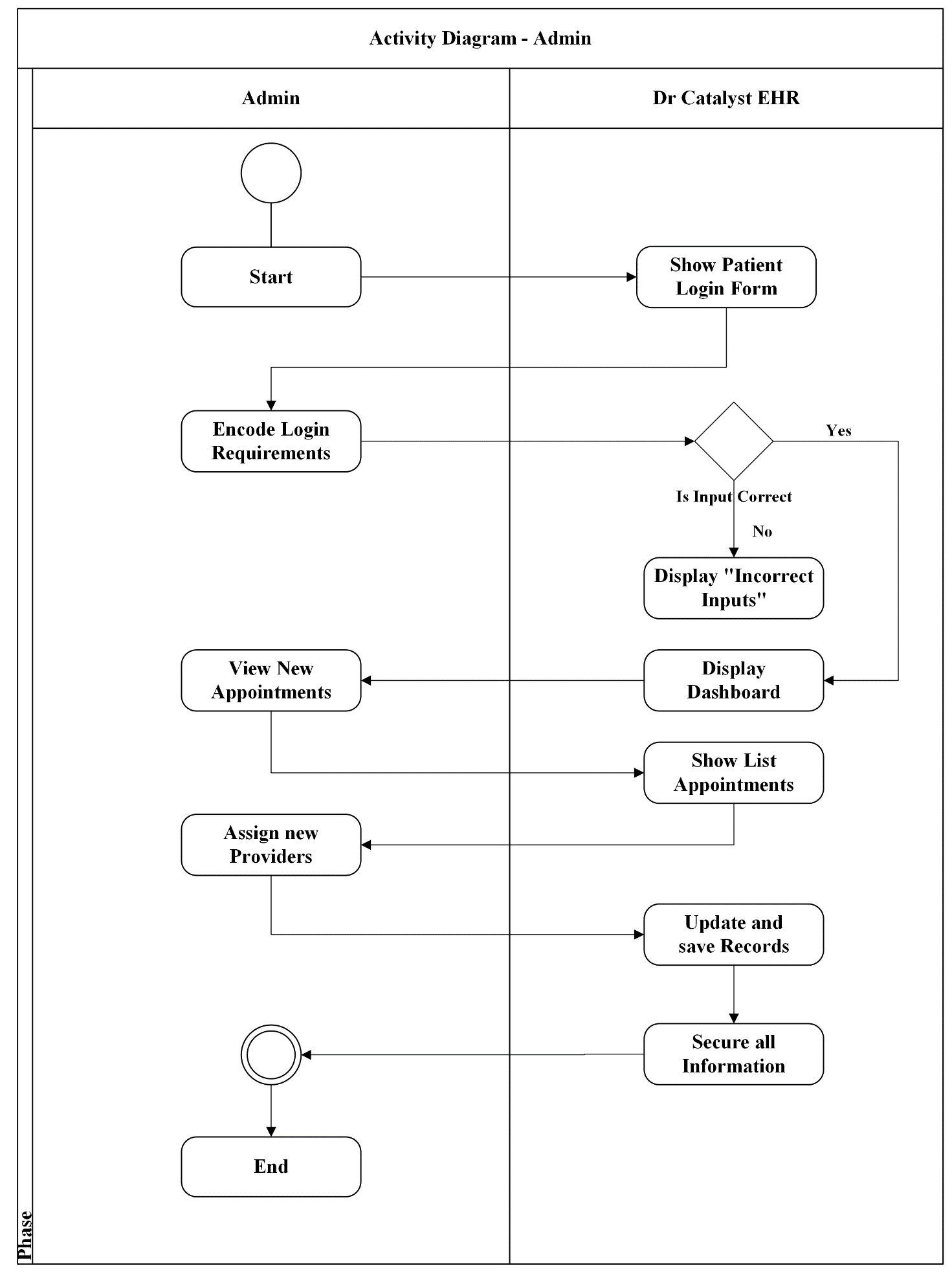
****

Fig. 3.5 Activitydiagram\_admin

Admin is able to see patient’s registered details through the system. Also patient related health details like allergies , vitals etc. are encoded and assigned a code with that. Once the patient details are encoded the admin will see the time schedule of the doctor and accordingly map the patient with the doctor in the schedule. also all the schedules can be updated and secured in the system. The complete flow is shown in the figure 3.5

**3.5 Data flow Diagram**

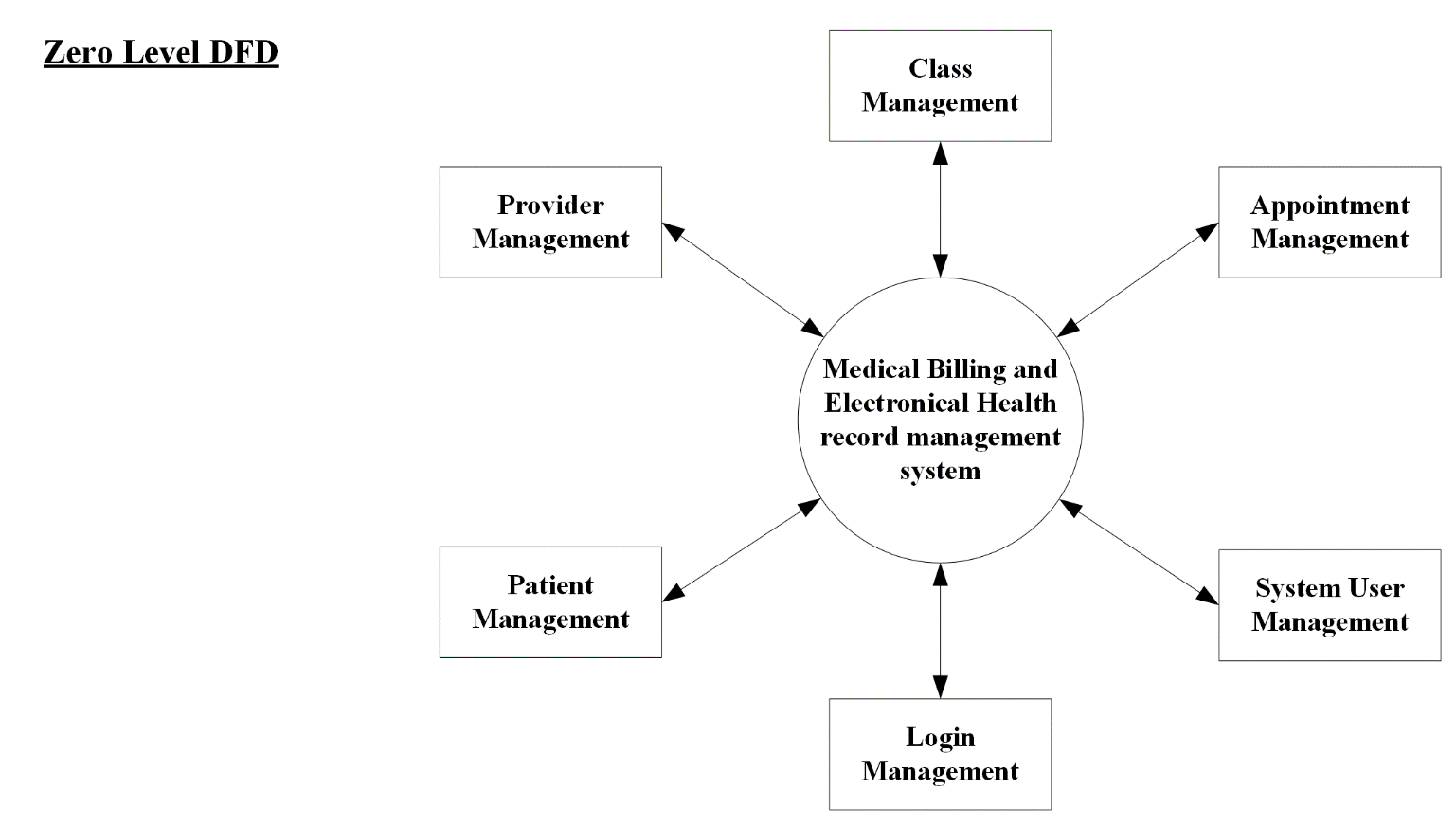
****

Fig. 3.6 Dataflowdiagram

Medical billing and electronic health record management system stores and manages the data of login credentials of patients, patient details such as name , address , allergy , payment, admission date and discharge date. it can also manage provider ( health insurance company ) details , such as company name , insurance amount, payment method etc . Appointment manages all the schedules of multiple patients so that there will be no clash. This is displayed in the figure 3.6

**3.6 Sequence Diagram**

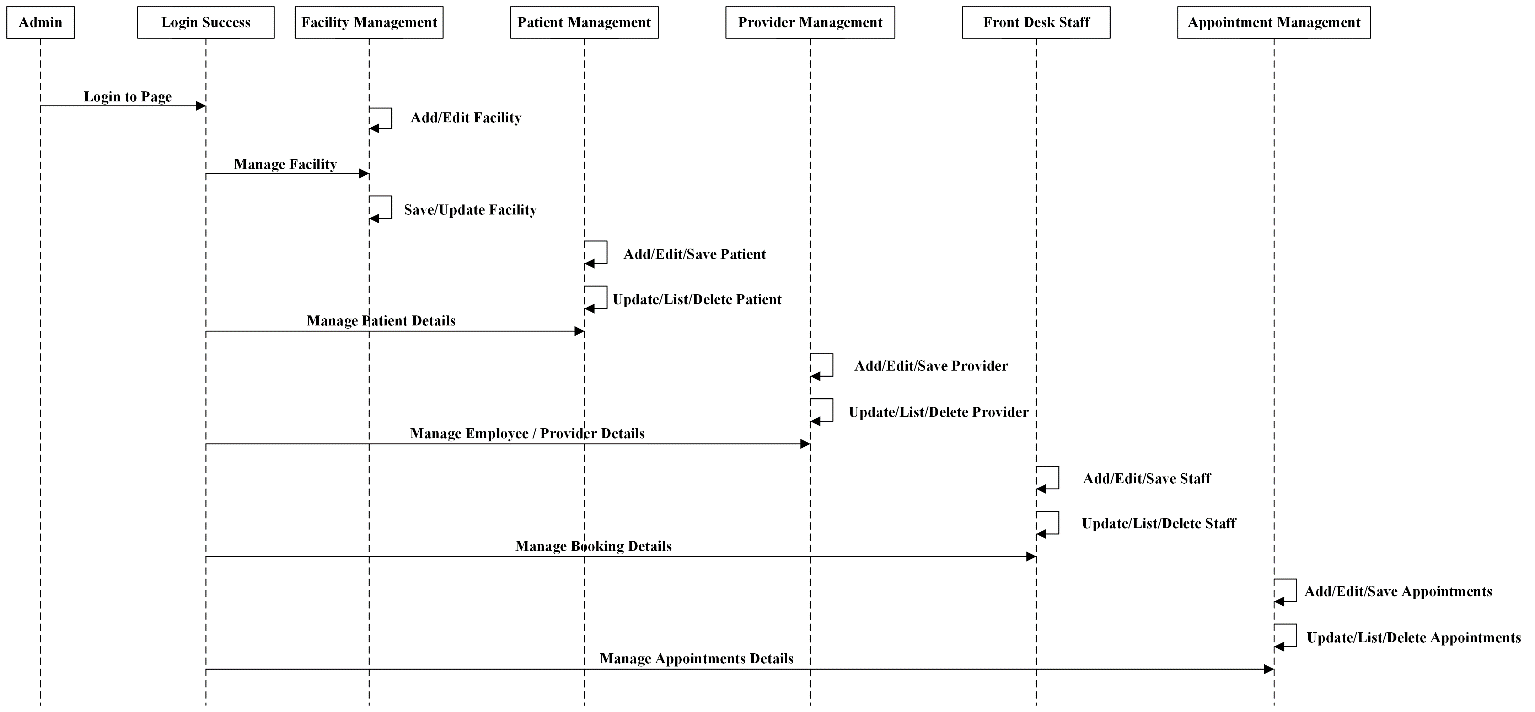
****

Fig. 3.7 Sequencediagram

The sequence diagram 3.7 indicates the things that an admin can perform. Once the admin has successfully logged in he/she will be able to perform edit or add or save operations in facility management, patient management, provider management , front desk staff and appointment management. Also all of this can be deleted by the admin.

**Chapter 4**

**Implementation**

**4.1 Demographics module**

**4.1.1 Front-end**

This is the part where front view of the demographics module of patient is developed this contains various patient details to be filled. The module is divided into four parts. First and second part is visible in the figure 4.1. First part contains information like first name, last name, Sex, DOB, Language preference, status, mantal Status, race, payment details etc.

The Second part is contact details of patient. Patient can have home, office, or other type of address. In that part also he can have primary address. In the address section the details like street, zip, city, state, country, phone number, fax number as per the figure 4.1.

# 

Fig. 4.1 Demographicswebview(1)

The third part is contact preference section, here primary fax number, phone number and address will be taken as per figure 4.3.

And at the last other details like employment of patient parent details, birth place, religion etc will be asked which is shown in the figure 4.2.

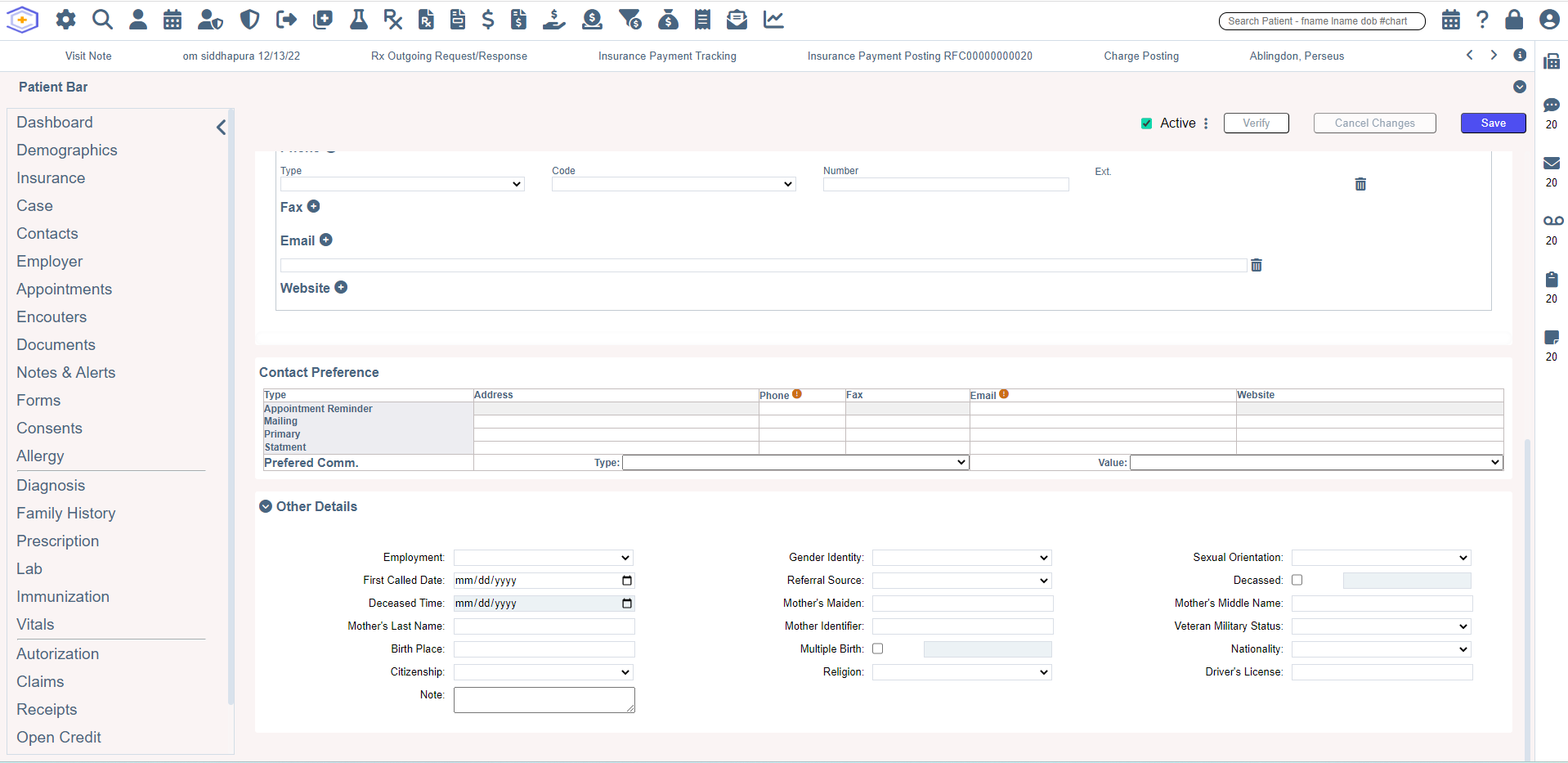


Fig. 4.2 Demographicswebview(2)

**4.1.2 Back-end**

**4.1.2.1 Operational Tables**

Total four APIs were created. All of them accesses patient, allergy and patient allergy tables as per the figure 4.3 . The figure 4.3 presents the part of database that will be required by the APIs.

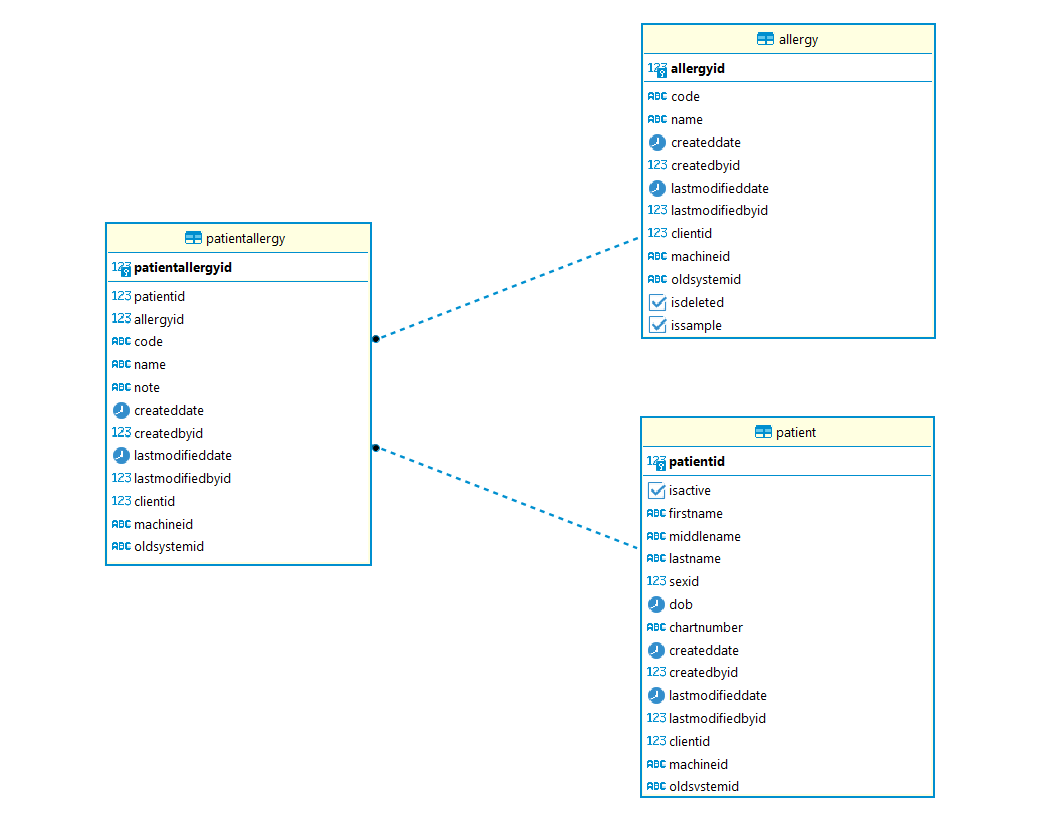


Fig. 4.3 operationaltables

**4.1.2.2 APIs**

**4.1.2.2.1 Create patient with allergy**

This API will create a new patient with all the provided details like firstname, lastname, sexid, dob etc. along with that if the optional allergy details are provided then that is also created in the database. The response contains patient details and allergies that were sent for creation which is visible in the figure 4.4.

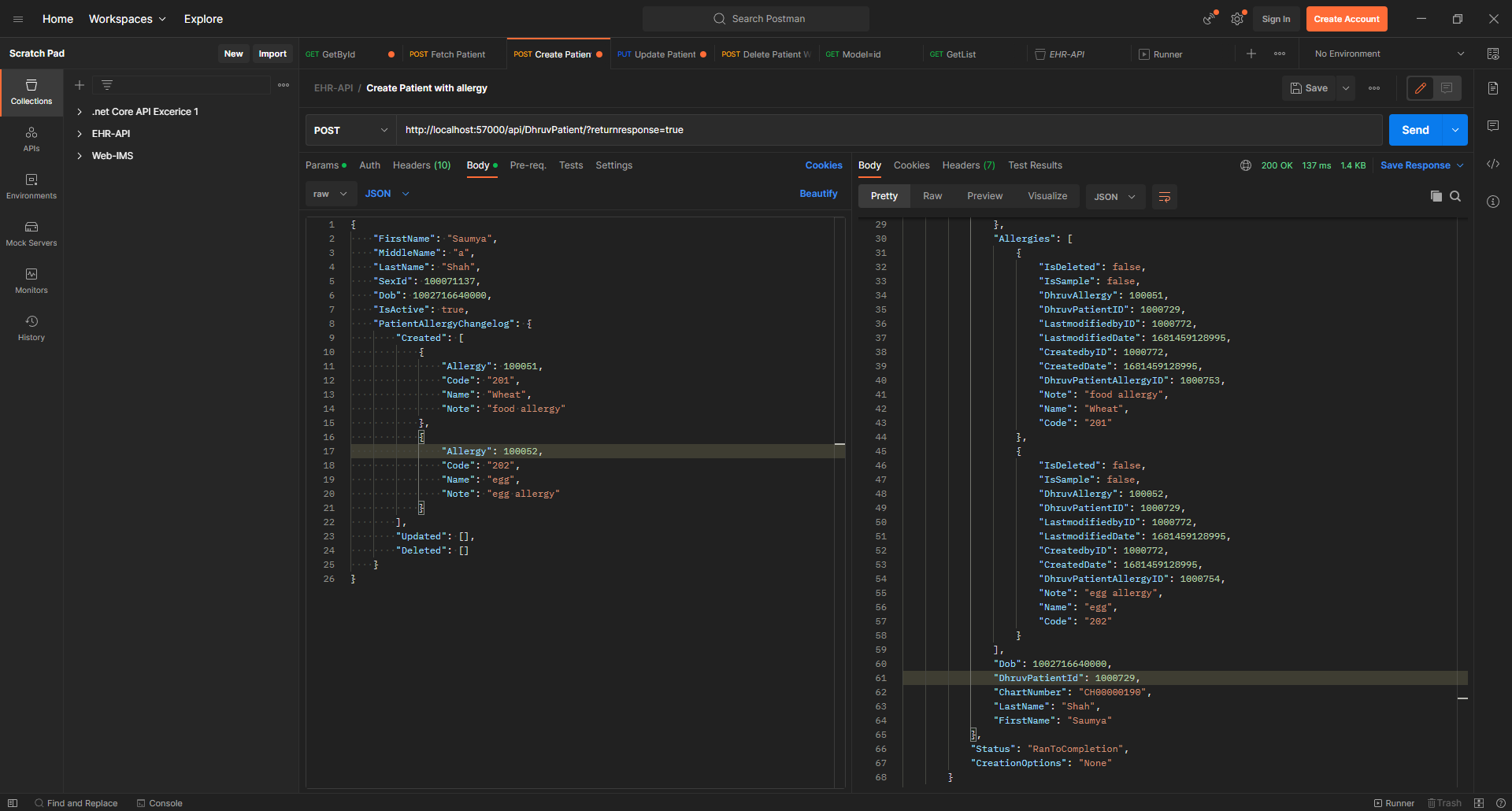


Fig. 4.4 Create patient with allergy

**4.1.2.2.2 Update patient with allergy**

As per the figure 4.5, this API will update old patient with all the provided details like firstname, lastname, sexid, dob etc. along with that if the optional allergy details are provided then the allergy will be deleted , created or updated based on the request. The response contains patient details and allergies that were sent for creation, updation or deletion.

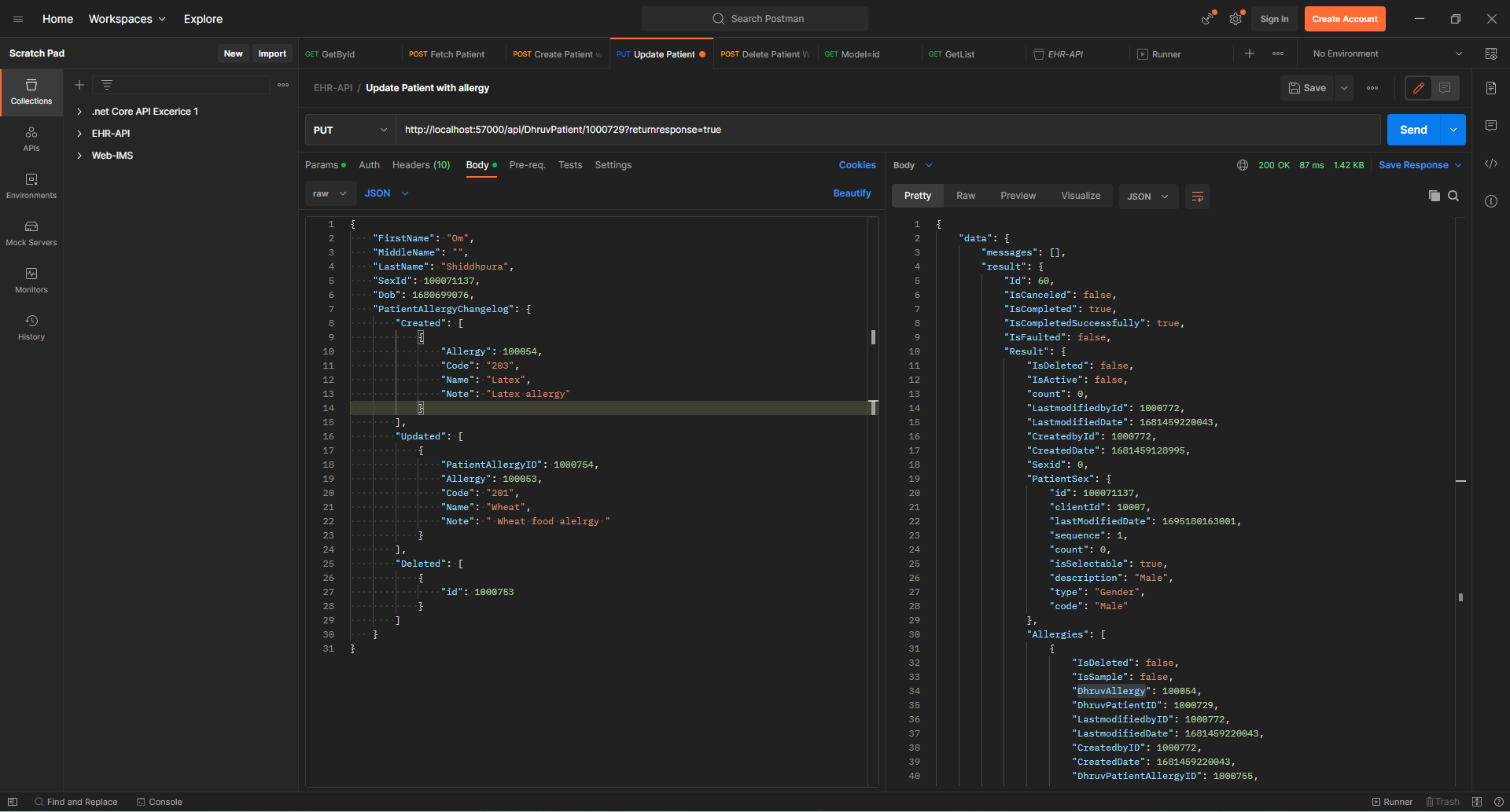


Fig 4.5 Update patient with allergy

**4.1.2.2.3 Get patient by id**

This API requires patientid as parameter along with that sex, allergy, createdby, getdb, patient object as optional flags. The main task of this API is to get patient’s information and corresponding allergies. All the optional flags are to get corresponding information only if they are true. If the patient object is true then all the information of the patient will be returned on success. The details are shown in the figure 4.6.

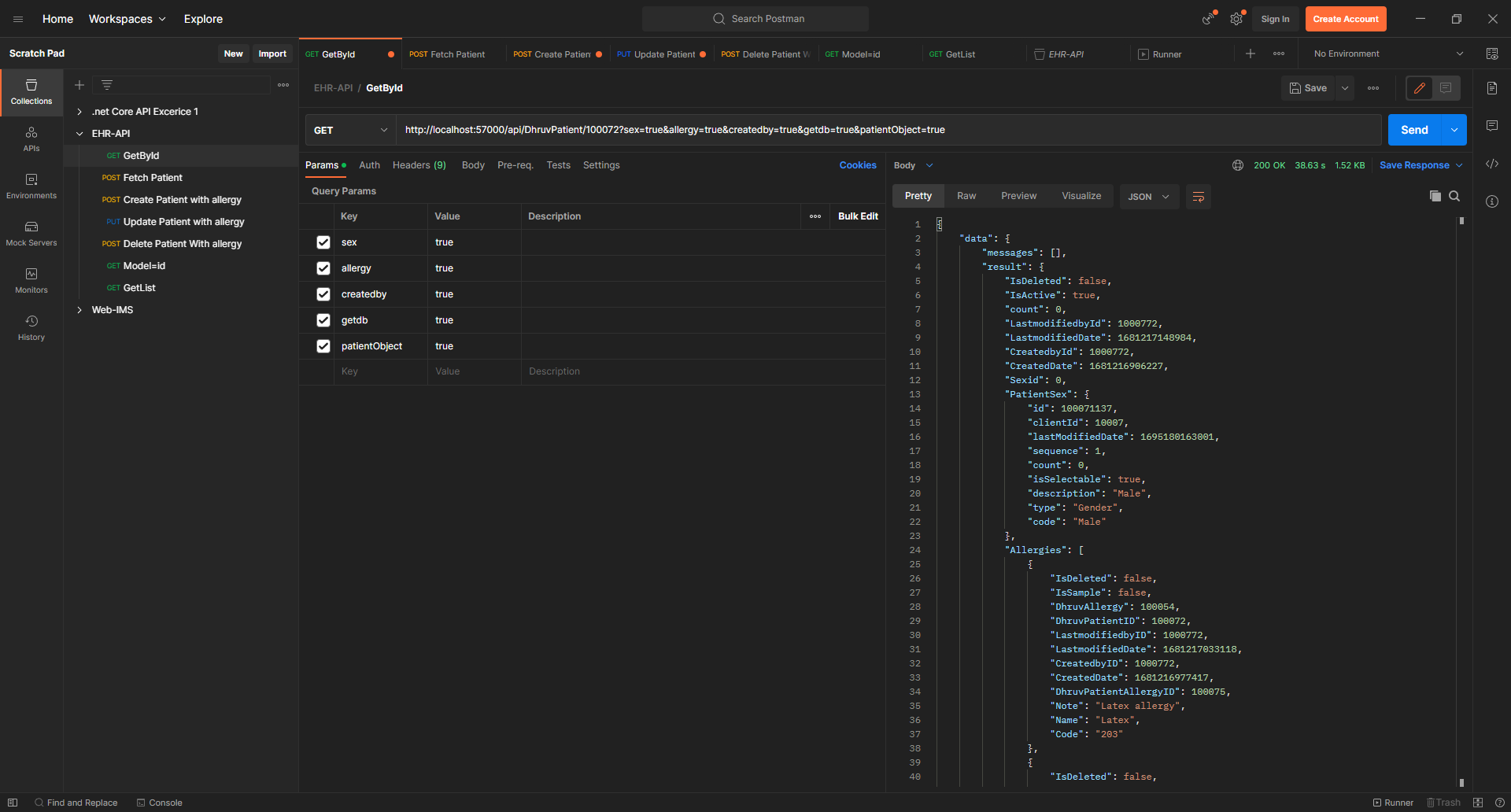


Fig. 4.6 Get patient details by id

**4.1.2.2.4 Delete patient with allergy**

This API requires patientid in the request body. The main task of this API is to delete patient’s information and corresponding allergies. On success the given patient related data will be deleted and success will be returned in the response which can be seen by the figure 4.7.

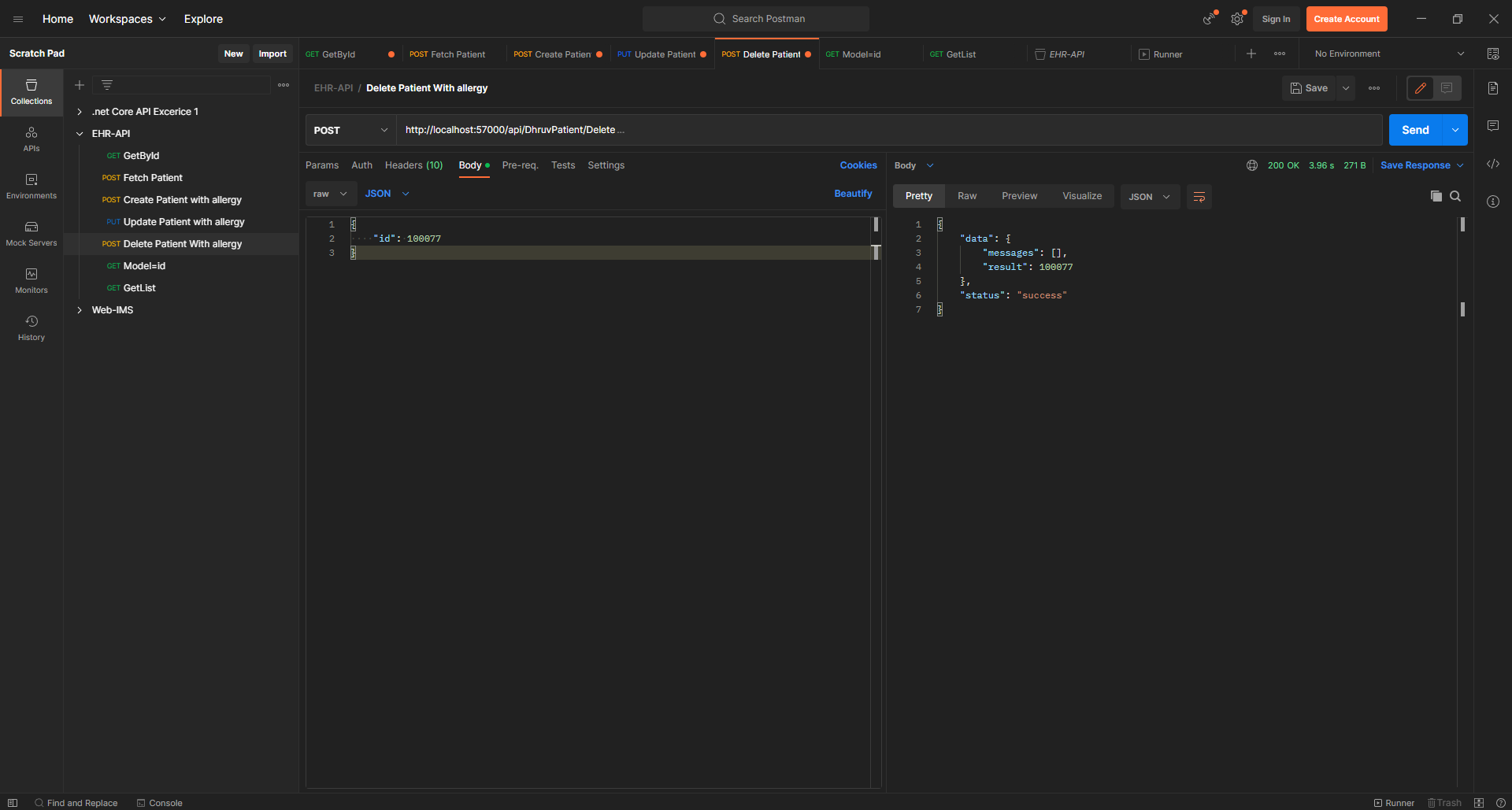


Fig. 4.7 Delete patient with allergy

**4.2 Check-In/Check Out module**

**4.2.1 Front-end**

The figure 4.8 presents the view of check in and check out of any patient in particular clinic during the visit of the patient. Whenever any patient arrives at the clinic for the appointment, he/she is added to the first section called appointment which is the first panel across the all panels as per the figure 4.8. Basic information of the patient can be viewed from here. Once the patient is entered in the clinic the arrival time is noted for the reference. From the first panel the patient can be shifted towards the other panel like waiting or in progress or rooms or checkout etc. as per other panels in the figure 4.8. At any instance some patient may be in arrival section and other patient in the waiting section some may with the provider in any room taking his/her treatment. Those who are completed with their treatment are shifted to checkout panel. The entire view can be visible in terms of panel and grid.

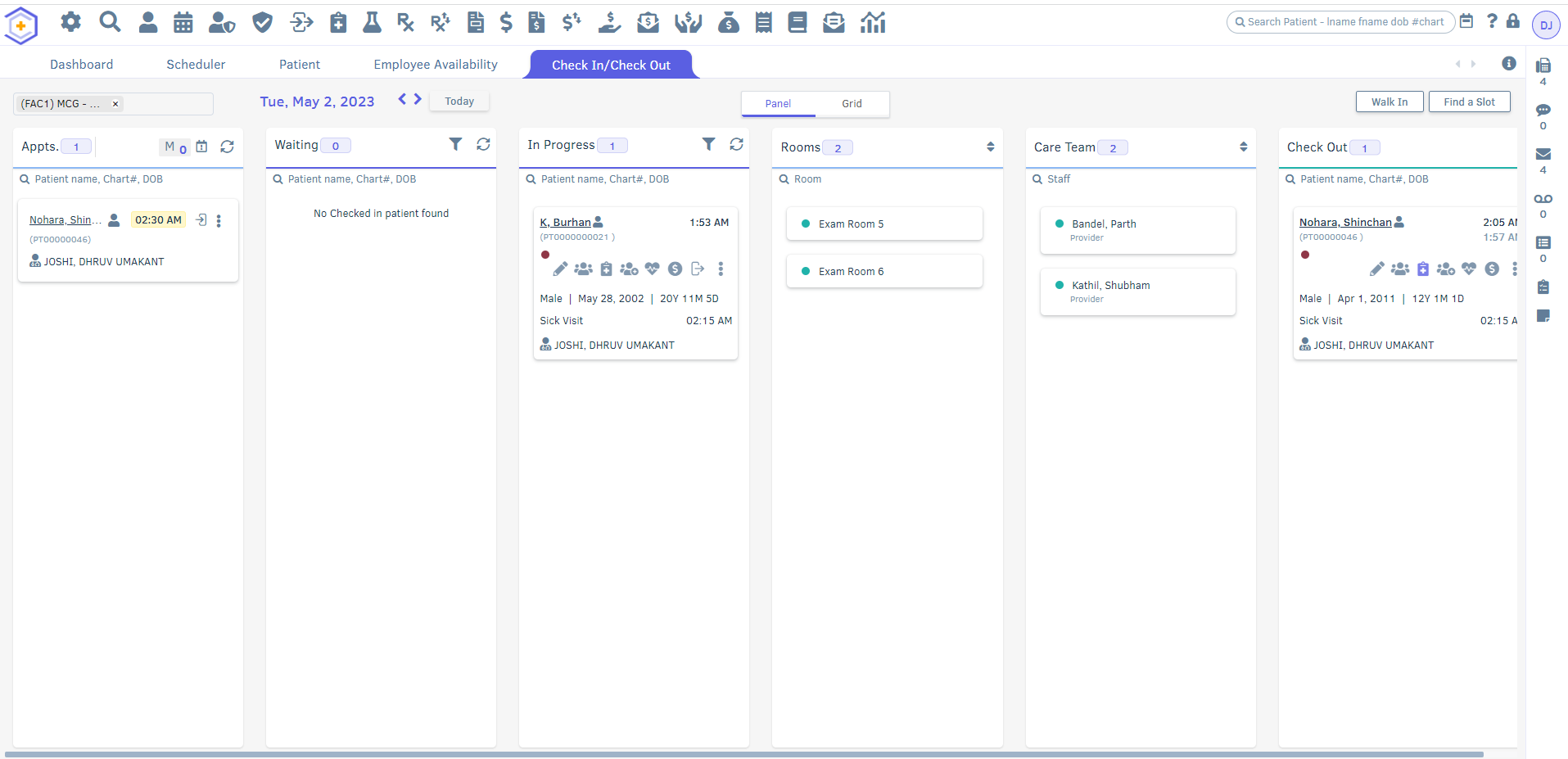


Fig. 4.8 CheckIn/CheckOut screen

The status of the patient can be controlled by the front desk manager by using the following menu. From here the manager can ask patients to fill forms as shown in figure 4.9. Various Insurance plans that the patient have is available to see. Also, the amount of time that patient has taken in the clinic and visit status of the patient like waiting or checkout and many more can be monitored and edited as shown in the figure 4.9.

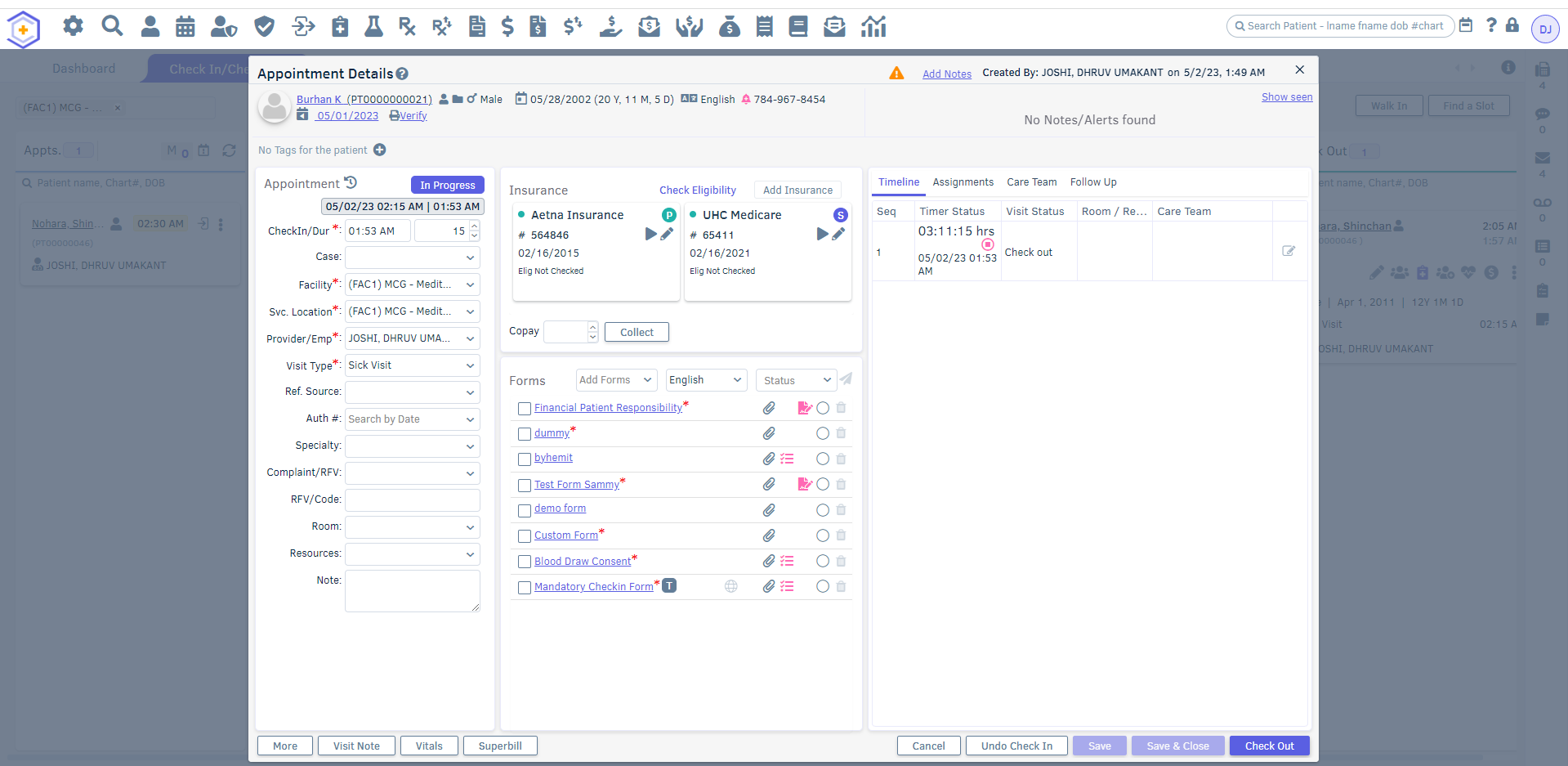


Fig. 4.9 Appointment Details

Appointment status can be tracked and edited for example the patient who has just arrived will be in patient arrived status which further can be changed to waiting area or in a room with MA. This screen is reached from the figure 4.9. From this screen the front desk manager can also assign a patient with the room number in which the treatment will be going on and the resources like room, waiting area, x-ray machine etc., this is displayed in the right section of the screen in the figure 4.10.

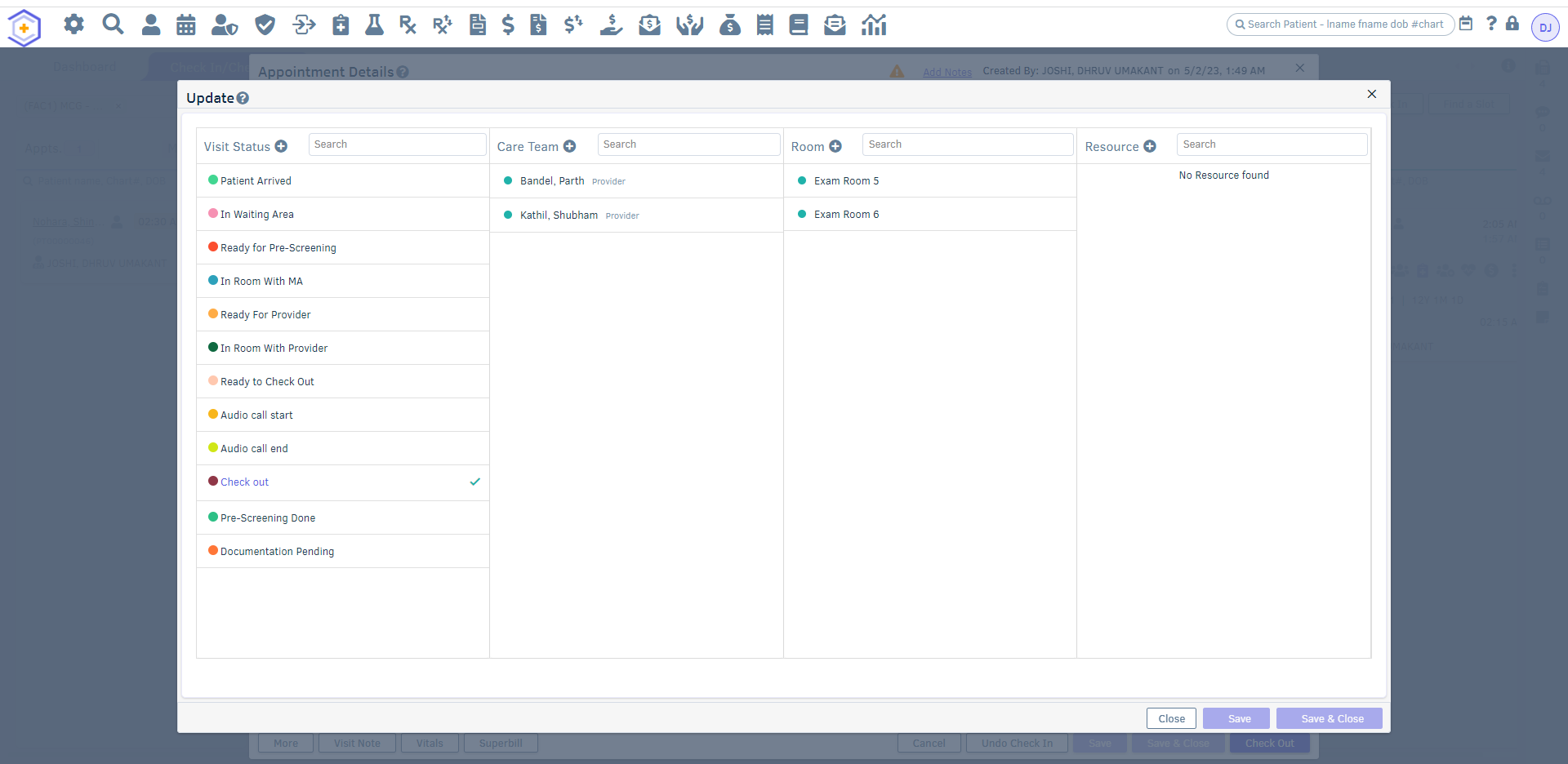


Fig. 4.10 Update Appointment

**4.3 Employee Availability module**

**4.3.1 Front-end**

The main functionality of this page is to schedule the presence of the employee so that the availability of the person can be known for the patient treatment related services where the employee can be doctor or lab assistant or front desk manager. The doctor can add a start date and end date, day name, template and days of the month. For example, a doctor will come on all Fridays during 06/01/2023 until a month during that period he/she will be providing a specific facility (FAC2) from 8 am to 5 am as shown in the second last row in the figure 4.11. The item can be edited or deleted. Template is nothing but the time structure in which the patient will be called to the doctor between the start time and end time. The entire record can be overridden and in other words updated.

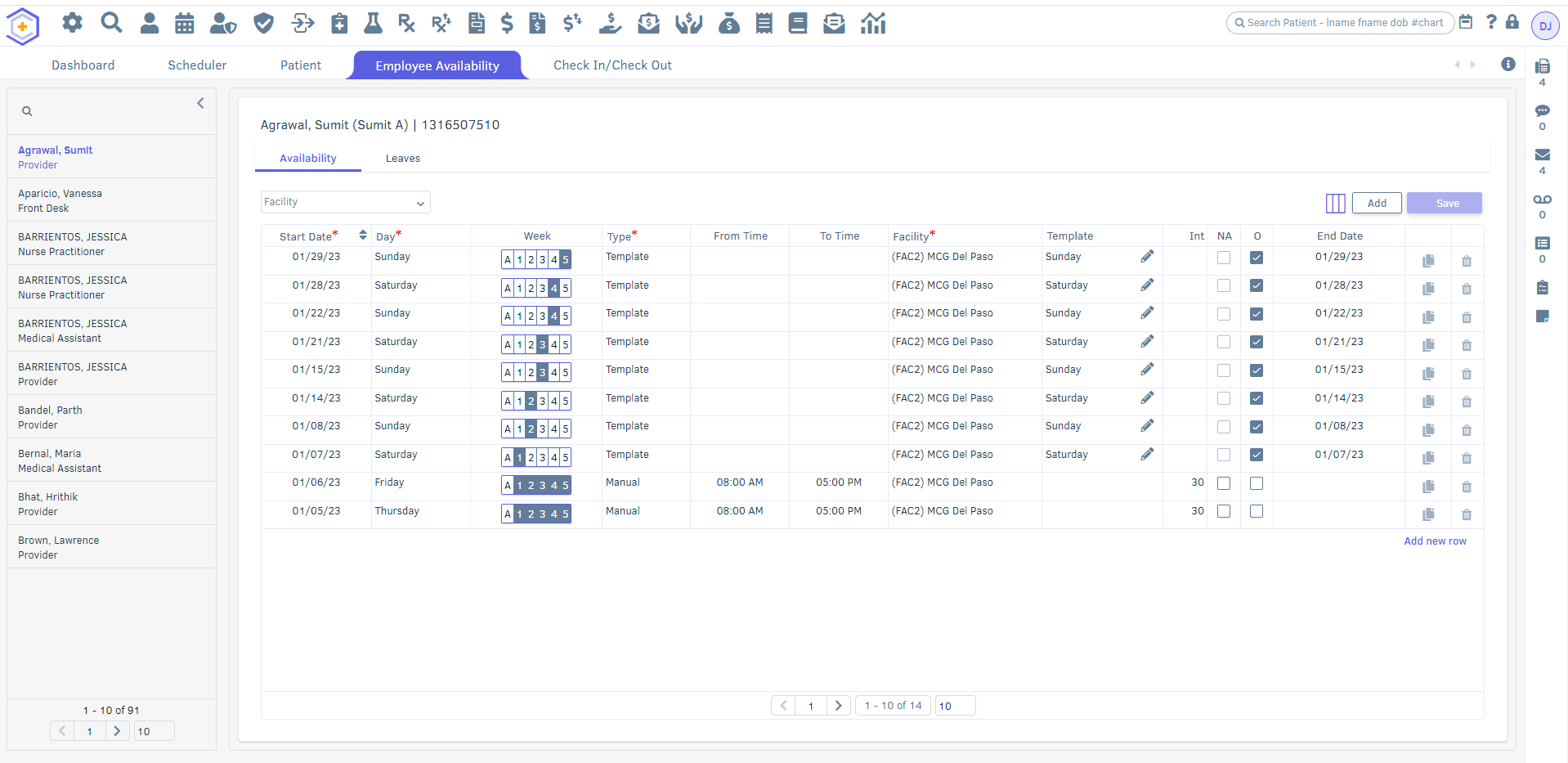


Fig. 4.11 Employee Availability

The figure 4.12 indicates the list of the remainders to the patient regarding health information. The reminder can be sent over text, email or call which can be deleted. And each reminder has a template.

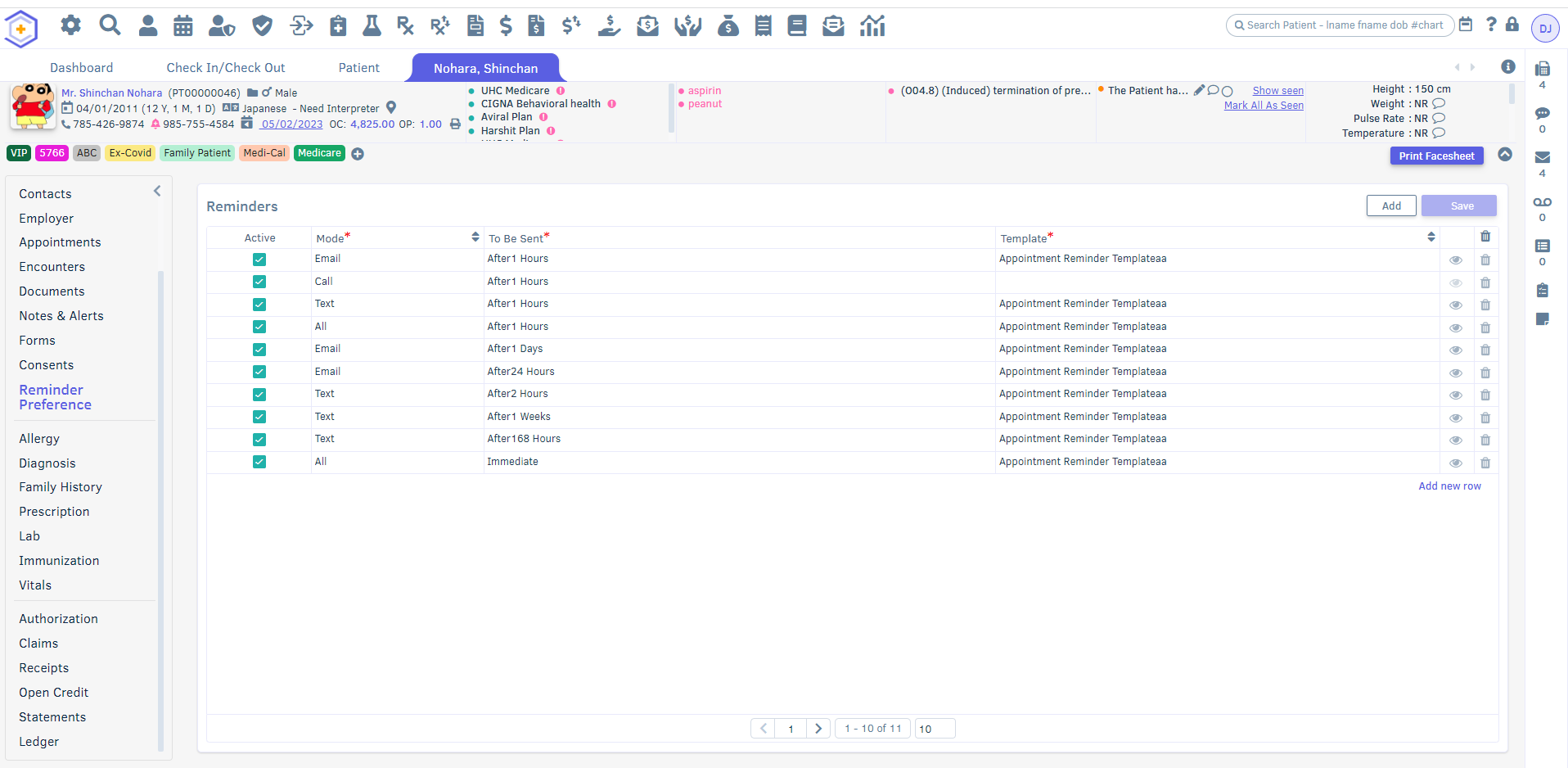


Fig. 4.12 Patient Reminders

# **Chapter-5**

**Conclusions and Future Scope**

**5.1 Conclusion**

Dr.Catalyst EHR is Web based software. It makes the life of providers (doctors) much easier by converting file-based systems to software-based systems. This is done by registering the entire patient in the IMS software and managing all the patient related information in the patient master module. Medical Insurance is mandatory in the US. To manage the patient insurance our system uses the patient insurance module to keep the records of the patient insurance and help to track all the claims. Thereafter, as the number of patients is huge, a search screen module is available in our software. This makes it easier for the user to search the patients, insurances, providers, etc. Instruction master is used to provide the facility of jotting down predefined instructions for a particular action. Authorization tracking module is used to track the claims that need to be sent to the insurance company.

**5.2 Limitation**

* This is web-based application so there will always come question about internet connectivity if there are no good internet connectivity problems will occur while using the application.
* It is totally server based so if there is a downtime going on server, then it may affect the system.
* While WebIMS offers integration with a range of third-party systems and applications, it may not integrate seamlessly with all systems or applications. This can lead to additional development and integration costs for healthcare organizations.
* Currently this application is only applicable in the USA and available in English and Spanish languages.
* This product has different features based on requirements from different clients so there are many bugs arising in the application.

**5.3 Future scope**

* The future plan is to make the application efficient and bug free.
* To make the application available in many languages.
* To make design UI friendly, easier and understandable by the healthcare practitioner.
* WebIMS may add more advanced analytics and reporting features to help clinicians and administrators analyze patient data and identify trends, risks, and opportunities for improvement.

**References:**

1. Java Script: https://www.javatpoint.com/javascript-tutorial
2. Java Script Image: https://www.freepnglogos.com/uploads/javascript-png/javascript-logo-transparent-logo-javascript-images-3.png
3. [.NET](https://dlib.org/) API: https://www.javatpoint.com/web-api#WebApi
4. [.NET](https://dlib.org/) API Image: https://www.learnopencv.com/histogram<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQtwymZBFk7aGmmDLZStaP6PNPj7K2TUTf08g&usqp=CAU>
5. Postgresql: https://www.tutorialspoint.com/postgresql/postgresql\_overview.htm
6. Postgresql Image: https://kinsta.com/wp-content/uploads/2022/02/postgres-logo.png
7. AWS: https://ajaymorya538.medium.com/what-is-cloud-what-is-aws-and-which-company-used-aws-69b63eff2d26
8. AWSImage: https://upload.wikimedia.org/wikipedia/commons/thumb/9/93/Amzon\_Web\_Services\_Logo.svg/1200px-Amazon\_Web\_Services\_Logo.svg.png
9. Postman: <https://www.javatpoint.com/postman>
10. Postman Image: https://assets.stickpng.com/images/62cc1b3a150d5de9a3dad5f7.png
11. Angular: https://www.tektutorialshub.com/angular/angular-introduction/
12. Angular Image: <https://repository-images.githubusercontent.com/24195339/87018c00-694b-11e9-8b5f-c34826306d36>